# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE - 1/16

# BY BOIAN MITOV

In the previous 2 articles, we introduced Visuino – a new graphical development environment for Arduino, and we demonstrated how you can use Delphi and the upcoming CommunicationLab from Mitov Software to communicate with Arduino, and get the data from its sensors into your applications. It surely is a lot of fun to use Visuino and Delphi to talk to Arduino, but it would be even bigger fun to be able to create your own components for Visuino and use them in your projects or share them with other developers, the same way as you do in Delphi.

When I started developing Visuino, one of my early goals was to develop it as open and expandable platform. Since the underlying OpenWire Studio already was developed around the concept of installable components, and packages, Visuino inherited this architecture.

From the beginning, all components in Visuino were parts of installable packages, and new components could be added, by simply installing new packages similar way as new components can be installed in Delphi.

As Visuino approached its release version, I also developed and released a Beta of the Visuino Components development SDK. The SDK is available for download from the Visuino Google+ community https://plus.google.com/

communities/116125623808250792822.

In this article you will learn how you can develop your own components for Visuino.

# START DEVELOPING VISUINO COMPONENTS

Before you start developing Visuino components, you will need to install Visuino and the Visuino components development SDK.

The SDK currently requires Delphi XE8 to be used for the component development. The typical Visuino installation will place the executable under C:\Program

Files\Mitov\Visuino or C:\Program Files
(x86)\Mitov\Visuino, depending on the
version of Windows you use.

The Visuino directory will contain the executable Visuino.exe . There is also a **Demos sub-directory**. There is another typically empty directory called **"Component Packages**". This is the directory where additional component packages can be installed.In addition to this folder structure, Visuino also instals **C++** Arduino files into "Mitov" sub-directory of the Arduino library folder. Typically the Arduino libraries folder is

**"My Documents\Arduino\libraries"**. Each Visuino component consists of 2 parts. The Arduino C++ code in the

"libraries\Mitov" directory, or another "libraries\\*" sub-directory, and a corresponding Delphi component in a package in the "Component Packages" directory. The C++ code is the one that compiles and executes in Arduino. The Delphi component is almost just an empty shell, that has some attributes needed to describe the capabilities of the C++ code. From that point of view, in most cases the Delphi code is just a simple description of the component. In some cases however, some more complex code generation logic can be implemented in the Delphi component, to control the way the Arduino code is generated.

The Components SDK after installation, creates its own folder structure. Currently, the typical installation will create "OpenWire" and "LabPacks" sub-folders under the Delphi main folder. In the "LabPacks\Mitov\Arduino\Examples" folder, there is an example of component package for Visuino.

In the C:\Program Files "LabPacks\Mitov\Arduino\ Examples\library" folder is where the C++ code for Arduino is located.

In the "LabPacks\Mitov\Arduino\ Examples\XE8" folder, is the VisuinoExampleProjectGroup.groupproj project group, containing the projects for the example Visuino package.

If you open the project group, you will discover that it contains 2 projects.

#### Visuino.ExamplePackage.dproj and Visuino.ExamplePackage.Design.dproj.

Visuino.ExamplePackage.dproj contains the demo components, and the Visuino.ExamplePackage.Design.dproj



# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE - 2/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



Visuino.ExamplePackage - RAD Studio 7	XE8 - Visuino.QuadraticFunctionExample	
<u>F</u> ile <u>E</u> dit <u>S</u> earch <u>V</u> iew Refact <u>o</u> r	r <u>Project</u> Run <u>C</u> omponent <u>T</u> ools C <u>a</u> stalia <u>W</u> indow <u>H</u> elp 🔇 Default Lay	out 🗸 🛱 🗗 👂 Search
in a n 👘 😰 - ei e r	δ Add to Project Shift+F11 32-bit Windows ▼	- 🖉 🔶 - 🔶 - 🥝
Structure 🕴 🗮 🕷	🖹 🐻 Remove from Project aticFunctionExample 🍯 < > 🛩 🕷	Visuino.ExamplePackage.dproj - Project Manager 🕴 🗮 🕷
🕀 🛅 Classes 👔	Add to Repository	× • • •
TVisuinoQuadraticFunctionE	Languages	
Protected	View Source	G 및 표   는 • 및   및 • <i>≅</i> •
A: Single	Add New Project	File
B: Single	Add Existing Project ples::QuadraticFunctionE	George Build Coningurations (Debug)
C: Single     Procedures	Se OA Metrics	e
🗄 💳 Uses	S QA Audits e = class ( TArduinoCommo-	
	🗊 Eormat Project Sources	🦾 🖹 Visuino.QuadraticFunctionExample.pas
	Modeling Support	🗄 🛅 Requires
	Compile Visuino.ExamplePackage Ctrl+F9	🗄 🛄 Visuino.ExamplePackage.Design.bpl 😑
	Build Visuino.ExamplePackage Shift+F9	🖶 🐄 Build Configurations (Debug)
	Syntax Check Visuino.ExamplePackage	⊕ ≝ Target Platforms (Win32)
• III •	Information for Visuino.ExamplePackage write FA;	Contains
Object Inspector 🕴 🕷	Compile All Projects	En Visuine Miteur Summles Desister nes
Properties	Build All Projects	CA Program Files (v96)) Embarcadoro) Studio) 16 0) LabDacks) Mitoù A
» Custom Build To	Paceurees and Images write FB;	C:\Program Files (xoo)\Embarcadero\studio\10.0\Eabracks\iviitov\A
Design Class	Dependencies	Visuino.Exampi   Model View   Data Explorer   Multi-Device Pr
File Name Visuino.Quadratic	Project Page Ontions Suggest a design time r	Tool Palette
Form Name	Deployment / Restrict the possible	Search
Full Path C:\Program Files (	Options Shift+Ctrl+F11 write FC;	Delphi Projects
	- end;	Delphi Projects   Delphi Files
	· · //	Other Files     Other Files
	<pre>procedure RegisterPkg();</pre>	C++Builder Projects   C++Builder Files
	□ implementation	Unit Test
	50 // You can compile the proj	ect group,
	by selecting [Project Bui	Id All Projects   .
		+ Web Documents
	<pre>procedure RegisterPkg();</pre>	C++Builder Projects   EMS
	- begin	Delphi Projects   EMS
Custom Build Tool	[	C++Builder Projects   IntraWeb
	< Þ	Delphi Projects   IntraWeb
	21: 31 Insert Code History	C++Builder Projects   DUnitX
🧑 🤌 🚺 Output	Visuino.ExamplePack	🔺 💽 👘 5:18 PM

After compiling the project group, in the

"LabPacks\Mitov\Arduino\Examples\XE8\Win32\Debug" will contain the compiled packages. To install the example components package, you can copy the \*.bpl files into the "Component Packages" directory of Visuino.

If you run Visuino, you will see that a new "Quadratic Function Example" component is available:



# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE - 3/16 **CREATE YOUR OWN COMPONENTS FOR VISUINO**



0

÷

Ξ



# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE - 4/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



# THIS TIME IF YOU TRY TO COMPILE THE CODE, THERE WILL BE NO ERRORS:



# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE - 5/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



Visuino.MyVisuinoPackage - RAD Studio XE8 - Visuino.MyVisuinoPackage.dproj File Edit Search View Refactor Project Run Component Tools Castalia Window Help 🔇 Default Layout - 6 7 Search 🗈 🗗 🔯 🍟 📽 🔻 🖶 🔲 🐻 🐻 🐌 🔸 🗊 🕶 🔢 🔳 🖳 📮 🖆 🔡 32-bit Windows 💌 - 🙋 💠 - 📦 - 😮 🗙 🗙 Visuino.MyVisuinoPackage.dproj - Project Manager Structure 🕈 🕷 Welcome Page 🂴 Visuino.MyVisuinoPackage 🔫 🚽 🗲 - 🔎 🏝 🗕 📴 👘 🗣 package Visuino.MyVisuinoPackage; 영 福 패 🗄 - 🚳 🛛 🐺 - 🥪 -{\$R \*.res} File {\$IFDEF IMPLICITBUILDING This IFDEF should not be used by 🔈 Proje {\$ALIGN 8} Add New Project... 🗄 [ 🖓 Visu (\$ASSERTIONS ON) Add Existing Project... {\$BOOLEVAL OFF} 🗄 🖏 B {\$DEBUGINFO OFF} ⊕--∰ Т Save Project Group {\$EXTENDEDSYNTAX ON} -- 💼 d Save Project Group As... {SIMPORTEDDATA ON} {\$IOCHECKS ON} Rename {\$LOCALSYMBOLS ON} Add to Version Control {\$LONGSTRINGS ON} {SOPENSTRINGS ON} {\$OPTIMIZATION OFF] {\$OVERFLOWCHECKS OFF} {\$RANGECHECKS OFF} Object Inspector **# X** {\$REFERENCEINFO ON} Properties {\$SAFEDIVIDE OFF} C:\Users\mitov\Documents\Embarcadero\Studio\Projects\ProjectGr {\$STACKFRAMES ON} Visuino.MyVisui... Model View Data Explorer Multi-Device Pr... {\$TYPEDADDRESS OFF} {\$VARSTRINGCHECKS ON} Tool Palette **X** 4 {\$WRITEABLECONST OFF} 🛐 🗸 📘 🔁 Search {\$MINENUMSIZE 1} Delphi Projects {\$IMAGEBASE \$400000} Ξ (SDEETNE DEBUG) 🗄 Delphi Projects | Delphi Files {\$ENDIF IMPLICITBUILDING} Other Files Add a second Delphi package to the project group: 🗄 C++Builder Projects | C++Builder Files Unit Test rtl; Delphi Projects | Multi-Device Projects • C++Builder Projects ) 🔶 🔳 Code History 8:44 Insert E C++Ruilder Projecte | Multi-Device Proje Messages # X X New Items -▲ · <sup>™</sup> C++Builder Projects Search 👘 ActiveX DLL 🛅 DataSnap Server 🛅 DUnitX 🛅 EMS Console Control Panel Dynamic-link FireMonkey MDI Multi-Device Application Application Library Metropolis... Application Application • Multi-Device Projects 🛅 WebBroker 🛅 WebServices n 🔺 🛅 Delphi Projects Package Resource DLL SDI VCL Forms Service VCL Metropolis 🛅 ActiveX Application UI Application Wizard Application Application 🛅 DataSnap Server 🛅 DUnitX EMS Save this one as Visuino.MyVisuinoPackage.Design. 🛅 IntraWeb Save the project group as MyVisuinoProjectGroup 🛅 Multi-Device Projects WebBroker 🛅 WebServices 🛅 IntraWeb -- 🛅 Other Files 🛄 Unit Test OK Cancel Help

# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE - 6/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



Open the project options for the Visuino.MyVisuinoPackage, select as Target "All configurations". In the Description settings page, type a description – as example "My Visuino Package", and set the package to be "Runtime only":

Project Options for Visuino.MyVisuin	oPackage.bpl (All platforms - All configurations)
Delphi Compiler     Compiling     With the second sec	Iarget:       All configurations - All platfo       ▲pply       Save         Description       My Visuino Package       Usage options       Build control         Usage options       Build control       @ Rebuild as needed       @ Explicit rebuild         @ Runtime only       @ Explicit rebuild       @ Explicit rebuild         Ubrary name       Library name       Library name         LIB suffix:       III Suffix:       III Suffix:
	OK Cancel Help
In the "Delphi Compiler' directory" to :.\\$(Pla And the "Search Path .\\$(Platform)\\$(Config);\$ \$(BDS)\LabPacks\Mitov form)\\$(Config)	<pre>' page set: "DCP output directory" and "Package output atform)\\$(Config) "" to: 5(BDS)\LabPacks\Mitov;\$(BDS)\LabPacks\Mitov\Arduino; D\XE8\\$(Platform)\\$(Config);\$(BDS)\LabPacks\Mitov\Arduino\XE8\\$(Platform))</pre>
Project Options for Visuino.MyVisuino	Package.bpl (All platforms - All configurations)
Delphi Compiler      Compiling	Target: All configurations - All platforms
Generating     Hints and Warnings     Hinking     Output - C/C++     Resource Compiler     Directories and Conditionals	Conditional defines         DCP output directory         -\\$(Platform)\\$(Config)         Framework search path, iOS Device - 32 bit, iOS         Package output directory         Search path         Search path         .\\$(Platform)\\$(Config)         Search path         .\\$(Platform)\\$(Config)
Build Events     Description     Version Info     Packages     Debugger	System library root path, iOS Device - 32 bit, iO Unit aliases Unit output directory Unit scope names System;Xml;Data;Datasnap;Web;Soap Use MSBuild externally to compile
Symbol Tables	
	OK Cancel Help

# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE - 7/16



**CREATE YOUR OWN COMPONENTS FOR VISUINO** х Project Options for Visuino.MyVisuinoPackage.Design.bpl (All platforms - All configurations) Delphi Compiler Target: All configurations - All platforms Apply... Save... Compiling Hints and Warnings Description Linking My Visuino Design Package Output - C/C++ A Resource Compiler Usage options Build control ..... Directories and Conditionals Build Events Designtime only Rebuild as needed Description Runtime only Explicit rebuild Version Info O Designtime and runtime · Packages ▲ · Debugger Library name Symbol Tables LIB prefix: Environment Block LIB suffix: LIB version: Do the same settings for the Visuino.MyVisuinoPackage.Design. You should set a different description, as example "My Visuino **Design Package":** OK Cancel Help 🚺 Visuino.MyVisuinoPackage.Design - RAD Studio XE8 - Visuino.MyVisuinoPackage.Design.dproj <u>C</u>ompile <u>B</u>uild File Edit Search View Refactor Project Run Component Tools Castalia Window Help 🥥 Default Layout - 6 Clean 🗈 🗗 🛐 🍟 💕 🗸 🔚 🔚 🐼 🐻 🐻 🕨 🔻 🖓 🕶 🔢 🔳 🖳 📮 🖆 🚼 32-bit Windows 💌 Fr<u>o</u>m Here Structure 👎 🗶 come Page 🛄 Visuino.MyVisuinoPackage 🎉 Visuino.MyVisuinoPackage.Desi 🔦 🕨 🗶 R<u>u</u>n ا 🔻 😹 🏝 👻 🗖 package Visuino.MyVisuinoPackage.Design; Run Without Debugging G 🖬 🔤 🗄 Build Soo<u>n</u>er Ctrl+Up {SR \*.res] File Build <u>L</u>ater Ctrl+Down {\$IFDEF IMPLICITBUILDING This IFDEF should not be used by 💠 MyVisuinoProj (SALIGN 8) Show in Explorer Uisuino.M {SASSERTIONS ON} {SBOOLEVAL OFF} 🗄 📅 Build Co Add... {\$DEBUGINFO OFF} 🗄 🥌 Target P Add New {\$EXTENDEDSYNTAX ON} - 🚞 Contains Add Reference... {SIMPORTEDDATA ON} 🗄 💼 Require: {\$IOCHECKS ON} Remove File... {\$LOCALSYMBOLS ON} 🗄 🚺 Visuino.My {\$LONGSTRINGS ON} Remove Project 🗄 🖏 Build Co {SOPENSTRINGS ON} 🗄 🥌 Target P Save (SOPTIMIZATION OFF) 💼 Contains Save As... {\$OVERFLOWCHECKS OFF} {\$RANGECHECKS OFF} 🗄 🚞 Requires Object Inspector **# X** Rename {\$REFERENCEINFO ON} Properties Add to Version Control {\$SAFEDIVIDE OFF} C:\Users\mitov\Do cts\New fold File Name JinoPackage.dproj {SSTACKFRAMES ON} Visuino.MyVisui.. Activate Device Pr... {\$TYPEDADDRESS OFF} Full Path C:\Users\mitov\Do {\$VARSTRINGCHECKS ON} View Source Ctrl+V Tool Palette **# X** {\$WRITEABLECONST OFF} Sort By 🔊 🕶 📘 🔎 s (\$MINENUMSTZE 1) Dependencies... 🗉 Delphi Project {\$IMAGEBASE \$400000} Modeling Support... {\$DEFINE DEBUG} 🗉 Delphi Project E {\$ENDIF IMPLICITBUILDING} Format Project Sources... Other Files {\$DESCRIPTION 'My Visuino Design Package'} • C++Builder Pr Options.. {\$RUNONLY} {\$IMPLICITBUILD ON} 🗉 Unit Test File Name Delphi Projects | Multi-Device Projects • C++Builder Projects ) 🔶 🔳 1: 9 Insert Modified Code History E.C., Ruilder Breierte I Multi Device Breierte Messages Right-click on the Visuino.MyVisuinoPackage, and select "View Source". Build Output 🚹 VisuinoExample 🛛 Visuino.MyVis... 😻 Visuino - Visu... 💿 Generated | Ar... 🔺 🎠 🏪 🌒 9:26 PM 🔒 Output 🔒 library 🔒 Visuino ß



# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE $\Omega/1$



CREATE YOUR OWN COMPONENTS FOR VISUINO					
Save the new unit as Visuino.MyComponent. In the unit add: unit Visuino.MyComponent;	Add a unit to the <b>Visuino . MyVisuinoPackage . Design ,</b> and save it as Visuino.MyPackage.Register. In this unit add:				
interface	unit Visuino, MyPackage, Register:				
<pre>uses System.Classes,Mitov.Design.Components, Mitov.Arduino.Types; type TVisuinoMyComponent =     class( TArduinoCommonAnalogFilter ) end; procedure RegisterPkg();</pre>	<pre>interface procedure Register(); implementation uses Visuino.MyComponent; procedure Register(); begin Visuino.MyComponent.RegisterPkg(); end;</pre>				
	end.				
<pre>procedure RegisterPkg(); begin     RegisterComponents(         [         TVisuinoMyComponent      ]     ); end; end.</pre>	When Visuino loads the Visuino.MyVisuinoPackage.Design package it will call the Register() method. The Register method will call Visuino.MyComponent.RegisterPkg(); RegisterPkg(); will call the RegisterComponents, and will pass array of components to be registered in the Visuino toolbar. If you install the packages in Visuino and run it. You will see that our new component is available:				
20 30					
	Image: State of the state o				

Issue Nr 5 2015 BLAISE PASCAL MAGAZINE

d 🍖 In

VisuinoMyComponent1

Out 🍋

# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE 10/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



The component however is not registered in any categories, does not have image, and does not have user friendly name. Lets start fixing this. Add Mitov.Arduino.Categories.Basic to the uses clause, and [Category( TArduinoMathFilterToolbarCategory)] attribute to the component:	To add the desired user friendly names, add Mitov.Attributes to the uses, and [CreateName('MyVisuinoComponent')], and [Name('My First Visuino Component' )] attributes to the component:			
uses System.Classes, Mitov.Design.Components, Mitov.Arduino.Types, Mitov.Arduino.Categories.Basic;	System.Classes, Mitov.Design.Components, Mitov.Arduino.Types, Mitov.Arduino.Categories.Basic, Mitov.Attributes;			
<pre>type [Category ( TArduinoMathFilterToolbarCategory ) TVisuinoMyComponent =     class ( TArduinoCommonAnalogFilter ) end;</pre>	<pre>type [Category(TArduinoMathFilterToolbarCategory)] [CreateName( 'MyVisuinoComponent' )] [Name( 'My First Visuino Component' )] TVisuinoMyComponent =     class( TArduinoCommonAnalogFilter ) end;</pre>			
This will place the component in the Arduino/Filters/Math subcategory, and the associated alternative categories.	You can also add image for the component. To do that, you need to create an empty Visuino.MyComponent.res resource file, and add 32x32 pixels PNG image as resource named <b>"TVISUINOMYCOMPONENT"</b> - upper case of our component name as declared in the code. I usually use Visual Studio to create and edit the resource files, but you can use any other resource editor.			
Visuino.MyComponent.res* - Microsoft Visual S $\nabla$ Quick Launch (Ctrl+Q) $P$ = $\Box$ X FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP Boian Mitrox <b>*</b> BM				
😋 🗸 🗇   🛅 🗶 🏣 🔛 🧬   🏷 - 🖓 -   🕨 Attach 🖒	* _ * J = - * - * - * - * - * - * - * - * * * *			
Visuino.MyComponent.res* 🕫 🗙 🗸	Properties 🝷 🕂 🗙			
E Visuino.MyComponent.res	Bitmap Node IBitmapRes -			
	Misc     (Name) Bitmap Node     Condition			
	Filename			
	ID "TVISUINOMYCOMPONENT"			
	ID Specifies the identifier of the bitmap resource.			
Ready				

Issue Nr 5 2015 BLAISE PASCAL MAGAZINE

# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE 11/16 **CREATE YOUR OWN COMPONENTS FOR VISUINO**



You can also open and explore the "Visuino.QuadraticFunctionExample.res" included in the SDK example. To use the resource in the component, add:

unit Visuino.MyComponent;

#### {\$R \*.res}

to the Visuino.MyComponent.pas file. Now if you rebuild, and deploy the package and run Visuino, you will see that the component has the new names and image, and appears in the right categories:

This is just the code to have an empty header file that has define preventing it from being included multiple times, and includes the Mitov.h header file that contains the Visuino base classes. Next you will create a namespace where you can put our component. This step is not really required, but is highly recommended to avoid conflicts of multiple components from multiple vendors with the same name. Use your own name, or company name for the namespace, or something else that is likely to be unique.



# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE 12/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



Now that you have a rudimentary Visuino This is enough to generate the proper code for component that you can test, go back to the Arduino, however the MyComponent inherits Delphi code, and instruct it how to generate code from Mitov::CommonEnableFilter which for the component. contains abstract method virtual void First you need to specify the header file that DoReceive( void \*\_Data ). You need to implement this method in your C++ needs to be included in the Arduino project when code, and this is where our data processing will be you use the component. In this case done. For now you can just do nothing with the Visuino MyComponent.h. You can do this by data, and send it to the output of the filter without adding [ArduinoInclude ( a change: 'Visuino MyComponent' )] attribute. Second you need to specify the namespace and class MyComponent : public Mitov::CommonEnableFilter the name of the C++ component. You can do this typedef Mitov::CommonEnableFilter inherited; by adding [ArduinoComponent( 'MyVisuinoComponents::MyComponent' protected: virtual void DoReceive( void \*\_Data ) ) 1 attribute: OutputPin.Notify( \_Data ); type [ArduinoInclude ( 'Visuino\_MyComponent' )] [Category ( TArduinoMathFilterToolbarCategory )] [ArduinoComponent( 'MyVisuinoComponents::MyComponent' )] [CreateName ( 'MyVisuinoComponent' )] [Name ( 'My First Visuino Component' )] TVisuinoMyComponent = class ( TArduinoCommonAnalogFilter DoReceive receives a pointer to the data, and it just calls OutputPin.Notify passing end: the same pointer to the data to be sent to the next filter in the chain. If you generate the code, and compile it in - O X 💿 Generated | Arduino 1.6.5 the Arduino IDE, it will File Edit Sketch Tools Help succeed: see figure at left. **0**-Your first component is • done, but it does not do Generated much. 11-\_\_\_\_\_ To be useful it needs to 11 perform some processing 11 Sketch Generated by Visuino over the data. 11 www.visuino.com Now you will implement 11 the same quadratic function 11-Ξ as implemented in the #define VISUINO ARDUINO UNO SDK's example, but you can easily #include <Mitov.h> implement on your own #include <OpenWire.h> any other function you can think of. #include <Visuino\_MyComponent.h> // Arduino Board DeclarationsDeclarations namespace Declarations Ł MyVisuinoComponents::MyComponent MyVisuinoComponent1; } // Declarations Done compiling. ÷ Global variables use 176 bytes (8%) of dynamic memory, leaving 1,872 bytes for local variables. Maximum is 2,048 bytes. Ξ Arduino Uno on COM1

# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE 13/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



For the quadratic function you will need 3 properties containing the 3 coefficients A, B and C. So lets declare them in the C++ code first:

```
class MyComponent : public Mitov::CommonEnableFilter
```

typedef Mitov::CommonEnableFilter inherited;
public:

float A,B,C;

{

Next, you need to initialize them with their default values in the constructor:



Finally you need to implement the proper computation in the DoReceive:

```
protected:
virtual void DoReceive( void *_Data )
{
    if( Enabled )
    {
      float AValue = *(float *)_Data;
      AValue = AValue * AValue * A + AValue * B + C;
      OutputPin.Notify( &AValue );
    }
    else
      OutputPin.Notify( _Data );
```

In this case if the component is Enabled it will obtain the floating point value from the \_Data pointer, use it for the calculation, and then send the result trough the OutputPin . If the component is not Enabled it will just send the data trough the OutputPin without changes. The C++ code for the component is ready. Now lets add the properties in the Delphi code.

First you need to declare the 3 fields that will hold the properties:

```
TVisuinoMyComponent = class( TArduinoCommonAnalogFilter )
protected
FA : Single; FB : Single; FC : Single;
end;
```

Next you need to declare the 3 properties, and specify the default values. The Visuino component framework will initialize the fields automatically with the default values:

# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE 14/16 **CREATE YOUR OWN COMPONENTS FOR VISUINO**



ΤV	<pre>/isuinoMyComponent = class( TArduinoCommonAnalogFilter ) protected FA : Single; FB : Single; FC : Single;</pre>	
	<pre>published [DefaultSingle(1.0)] property A : Single read FA write FA; [DefaultSingle(2.0)] property B : Single read FB write FB; [DefaultSingle(3.0)] property C : Single read FC write FC;</pre>	Your component is ready to be used. If you deploy the compiled *.bpl files to the "Component Packages" sub-directory of Visuino, and run Visuino, you will have the
	end;	you can edit them:

hem: Visuino - Visual Arduino Programming Beta 41 (MyTest1) - 0 **-** X <u>File Edit View Arduino H</u>elp 🔜 🕪 - 🔜 🛛 🕅 🖛 - 🍽 - Zoom: 100% - 🔍 🔍 🔍 -X 6 8 🔶 💽 - 🔄 🐂 🐂 **—** ₽. 🧊 Properties 🖓 🐮 🐮 📑 ø A Miscellaneous 🙀 MyVisuinoComponent1 A B Out 🍖 n In **H** 2 k C k Enabled Name 8 ✓ True MyVisuinoCompo ш

# Here is example Visuino connection diagram that you can use to test the new component:



# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE 15/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



Your component is ready, tested and operational.

There are some final improvements that you can do. At the moment the end user of the component can set any value for the A, B and C parameters. Sometimes you want to limit that range. You can limit the range of a property by adding the ValueRange attribute, as example like this:

```
[DefaultSingle( 3.0 )]
[ValueRange( -1000, 1000 )]
property C : Single read FC write FC;
```

You can also add a suggested smaller range that will be used in the Visuino property editor when showing the in-place track-bar value editor, by using the DesignRange attribute like this:

```
[DefaultSingle( 3.0 )]
[DesignRange( -100, 100 )]
[ValueRange( -1000, 1000 )]
property C : Single read FC write FC;
```

If you add those attributes, rebuild and deploy the package, in Visuino the in-place editor will offer only the -100 to 100 range:

# ARDUINO: THE VISUINO PROJECT - PART 3 PAGE 16/16 CREATE YOUR OWN COMPONENTS FOR VISUINO



Any attempt to also enter manually value outside the -1000 to 1000 range will fail.

If the DesignRange attribute is not present, and ValueRange is present, the track-bar will use the ValueRange instead. If none of them is present, the track-bar will offer the full range of floating point values.

### CONCLUSION

In this article you learned the basics of creating your own Visuino components. As you have seen, it is very easy, and can be done by almost anyone, even with limited programming knowledge.

The component we demonstrated is very simple, and yet already very useful. With the component SDK you can create much more complex and advanced components, with many more features. All components included in Visuino are written using this SDK, and you can see the power they offer. It is not possible in a single article to cover all aspects of Visuino component development, but this is a very good starting point. We are working hard to provide more resources and information on Visuino component development, and I hope you all will have many joyful hours playing with Visuino and creating your own components for it.

In the next Visuino article we will show you how you can connect Delphi applications and Arduino boards over internet, and how you can make different Arduino boards talk to each other. You will be entering the exciting world of "Internet of Things"!

